An Enhanced Dbscan Algorithm to Cluster Web usage Data using Rough Sets and Upper Approximations

K. Santhisree¹, A. Damodaram² & SV Appaji³

¹Department of Compterscience, JNT University, Hyderabad, India
²Department of Computer Science, Arora College of Engineering, India
¹kakara_2006@yahoo.co.in, ²damodaram@rediffmail.com, ³appaji_sv@rediffmail.com

ABSTRACT

Web usage mining is the application of data mining techniques to web log data repositories. It is used in finding the user access patterns from web access log. User page visits are sequential in nature. In this paper we presented new rough set Dbscan clustering algorithm which identifies the behavior of the users page visits, order of occurrence of visits. Web data Clusters are formed using the rough set Similarity Upper Approximations. We present the experimental results on MSNBC web navigation dataset. And proved rough set Dbscan clustering has better efficiency and performance compared to rough set agglomerative clustering.

1. INTRODUCTION

Clustering is of prime importance in data analysis, machine learning and statistics. It is defines as the process of grouping N item sets into distinct clusters based on similarity or distance function. A good clustering technique may yield clusters thus have high inter cluster and low intra cluster distance. (Pradeep Kumar and bapi, 2007) The objective of clustering is to maximize the similarity of the data points within each cluster and maximize dissimilarity across clusters.

2. LITERATURE REVIEW

2.1. Sequential Data

A sequence is an ordered list of items. A sequence S is denoted as <s₁, s₂, ..., sₙ> where s₁, s₂, s₃...sn are called the item sets in the sequence S an item can occur at multiple times in a sequence. the number of occurrences of an item in a sequence is called the length of the sequence. A sequence with length 1 is called the 1-sequence. The problem of mining sequential patterns was first introduced by Agarwal an Srikanth[1]. In order to find he patterns in the sequences it is necessary to not to look at the items contained in the sequences but also the order of their occurrence. In order to find patterns in sequences, it is necessary to not only look at the items contained but also on the order of their occurrences.

2.2. Similarity, Sequence Similarity

In data mining applications, we are given with unlabelled data (IJDWM 2007 pradeep, Bapi)) and we have to group them based on the similarity measure. Similarity is a function S with nonnegative real values defined on the Cartesian product X × X of a set X. It is called a metric on X if for every x, y, z, the following properties has to be satisfied by S.

1. Non-negativity: S(x, y) ≥ 0.
2. Symmetry: S(x, y) = S(y, x).

A new measure called the sequence and the set similarity measure S3M introduced which consists of two parts. One that quantifies the composition of the sequence (set similarity) and the other that quantifies the sequential nature (sequence similarity) sequence similarity quantifies the similarity on the order of occurrence of item sets within two sequences. Length of the longest common subsequence (LLCS) with respect to the length of the longest sequence determines the sequence similarity aspect across two sequences. consider two sequences X and Y, the sequence similarity is measured as (Pradeep and P.R krishna, 2007) set similarity is defined as the ratio to the number of common item sets and the number of unique item sets in two sequences.

Thus, S3M measure for two sequences A and B is given by

$$S3M(X, Y) = \frac{p * LLCS(X, Y) + q * |X \cap Y|}{Max(|X|, |Y|)}$$

Here, p + q = 1 and p, q ≥ 0. p and q determine the relative weights to be given for order of occurrence (sequence similarity) and to content (set similarity), respectively.
3. WEB USAGE DATA MINING

Web mining is the use of Data mining techniques to extract information from web documents and services. Web mining is decomposed into three sub tasks[7]. One important use of clustering in web usage mining is finding the groups which share common interests and behavior by analyzing the data collected in the web servers. This study contributes the topic clustering of web usage data and shows the interests and behaviors of the various user visits.

4. ALGORITHM

Proposed new Rough set DBscan clustering

**Input:** n

- T: A set of n transactions $\epsilon$ U
- Threshold $d \in [0, 1]$
- Relative similarity $r \in [0, 1]$
- Epsilon $\epsilon [0, 1]$
- Minpts: number of Neighborhood points

**Output:**
- Cluster scheme C

**Begin**

**Step 1:** Construct the similarity matrix using $S3M$ measure. Using definition 1

**Step 2:** select all points from D that satisfy the Eps and Minpts

- $C = 0$
  - for each unvisited point $P$ in dataset D
  - mark $P$ as visited
  - $N = \text{get Neighbors} (P, \epsilon)$
  - if size of $N' < \text{MinPts}$
    - mark $P$ as NOISE
  - else
    - begin
      - $C = \text{next cluster}$
      - mark $P$ as visited
    - end
  - add $P$ to cluster $C$
  - for each point $P'$ in $N$
    - if $P'$ is not visited
      - mark $P'$ as visited

**Step 3:** Return $C$

**Step 4:** For all $Ti \in U$ Compute $Si = R(Ti)$

Using definition 2 for given threshold $d$.

**Step 5:** Next compute the constrained-similarity upper Approximations $S_j$ for relative similarity $r$ using definition 3

- if $Si = S_j$
  - endif

**Step 6:** Repeat step 3 until $U \neq \emptyset$;

Return $D$.

**End**

5. EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Sample Sequences</th>
<th>Order of user page visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3 2 2 4 2 2 2 3 3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>6 7 7 6 6 8 8 8 8</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>6 9 4 4 4 0 3 1 0 5 1 0 4 4 4</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>12 12</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>11</td>
</tr>
</tbody>
</table>

**Description of the Dataset**

We collected the data from the UCI dataset repository (http://www.ics.uci.edu) that consists of sever logs from msnbc.com for the month of September 1998. Each sequence corresponds to page views of a user during that 24 hour period. Each sequence in the dataset corresponds to the page views of a user during that twenty four r hour period. Each event in the sequence corresponds to a users request for a page.

Each category is associated—in order—with an integer starting with “1”. For example, “FrontPage” is associated with 1, “news” with 2, and “tech” with 3. Each row below “% Sequences:” describes the hits in order of a single user. For example, the first user hits “FrontPage” twice, and the second user hits “news” once.

Output: From the above algorithm, the first similarity upper approximation at threshold value 0.5 is given by

\[
\begin{align*}
R(T1) &= \{T1, T5, T7, T11, T13\}, R(T2) = \{T2\}, R(T3) = \{T3\}, \\
R(T4) &= \{T4\}, R(T5) = \{T1, T5, T7, T11, T13\}, R(T6) = \{T6, T8\}, \\
R(T7) &= \{T1, T7, T11, T13\}, R(T8) = \{T6, T8\}, R(T9) = \{T9\}, \\
R(T10) &= \{T10\} \\
R(T11) &= \{T1, T5, T7, T11, T13\}, \\
R(T12) &= \{T1, T5, T7, T11, T13\}
\end{align*}
\]

The second upper approximations with the proposed value 0.5 are

\[
\begin{align*}
RR(T1) &= \{T1, T5, T7, T11, T13\}, RR(T2) = \{T2\}, RR(T3) = \{T3\}, \\
RR(T4) &= \{T4\}, RR(T5) = \{T1, T5, T7, T11, T13\}, RR(T6) = \{T6, T8\}, RR(T7) = \{T1, T5, T7, T11, T13\}, RR(T8) = \{T6, T8\}, \\
RR(T9) &= \{T9\}, RR(T10) = \{T10\}, RR(T11) = \{T1, T5, T7, T11, T13\}, RR(T12) = \{T1, T5, T7, T11, T13\}
\end{align*}
\]

The third upper approximations

\[
\begin{align*}
RRR(T1) &= \{T1, T5, T7, T11, T13\}, RRR(T2) = \{T2\}, RRR(T3) = \{T3\}, \\
RRR(T4) &= \{T4\}, RRR(T5) = \{T1, T5, T7, T11, T13\}, RRR(T6) = \{T6, T8\}, RRR(T7) = \{T1, T5, T7, T11, T13\}, RRR(T8) = \{T6, T8\}, \\
RRR(T9) &= \{T9\}, RRR(T10) = \{T10\}, RRR(T11) = \{T1, T5, T7, T11, T13\}, \\
RRR(T12) &= \{T1, T5, T7, T11, T13\}
\end{align*}
\]

The similarity upper approximations for

\[
\begin{align*}
C1 &= \{T1, T5, T7, T11, T13\}, C2 = \{T2\}, C3 = \{T3\}, C4 = \{T4\}, \\
C5 &= \{T1, T5, T7, T11, T13\}, C6 = \{T6, T8\}, C7 = \{T1, T5, T7, T11, T13\}, C8 = \{T6, T8\}, C9 = \{T9\}, \\
C10 &= \{T10\}, C11 &= \{T1, T5, T7, T11, T13\}, C12 = \{T12\}, C13 = \{T1, T5, T7, T11, T13\}
\end{align*}
\]

Epsilon value=0.2 and Epsilon neighbour is 3

C1={T5, T7, T13}, C2={T2}, C3={T3}, C4={T4}, C5={T1}, C6={T8}, \\
C7={T7}, C8={T6}, C9={T9}, C10={T10}, C11={T11}, C12={T12}, C13={T1}

T1 is a border point which results in the clusters c13 and c5.

6. COMPARISON OF EXPERIMENTAL RESULTS

The clusters formed using rough set agglomerative clustering are

\[
\begin{align*}
C1 &= \{T1, T5, T7, T11, T13\}, C2 = \{T2\}, C3 = \{T3\}, C4 = \{T4\}, \\
C5 &= \{T1, T5, T7, T11, T13\}, C6 = \{T6, T8\}, C7 = \{T7\}, C8 = \{T6, T8\}, C9 = \{T9\}, C10 = \{T10\}, \\
C11 &= \{T1, T5, T7, T11, T13\}, C12 = \{T12\}, C13 = \{T1, T5, T7, T11, T13\}
\end{align*}
\]

The clusters formed after applying DBscan clustering algorithm on the 13 sets are Epsilon value=0.2 and Epsilon neighbour is 3

\[
\begin{align*}
C1 &= \{T5, T7, T13\}, C2 = \{T2\}, C3 = \{T3\}, C4 = \{T4\}, \\
C5 &= \{T1, T5, T7, T11, T13\}, C6 = \{T6, T8\}, C7 = \{T7\}, C8 = \{T6, T8\}, C9 = \{T9\}, C10 = \{T10\}, \\
C11 &= \{T1, T5, T7, T11, T13\}, C12 = \{T12\}, C13 = \{T1, T5, T7, T11, T13\}
\end{align*}
\]

T1 is a border point which results in the clusters c13 and c5.

7. CONCLUSIONS

In this paper we developed a new rough set DBscan clustering algorithm and presented experimental results on msnbc.com which is useful in finding the user access patterns and the order of visits of the hyperlinks of the each user and the inter cluster similarity among the clusters. The rough set DBscan clustering algorithm is efficient when compared to the rough set agglomerative clustering. As in rough set agglomerative clustering the elements can be present in more than one cluster(soft clustering), where as in our proposed algorithm rough set DBscan algorithm (hard clustering), the elements will not occur in other clusters.

REFERENCES